

Biological Computation 20.181

Homework 3

In this assignment, you will get more experience writing functions that act on the tree data structure we outlined in class.

Tree arithmetic. In this exercise, you will compute the average distance from the root of a tree to its leaves. In the last homework, you were given the task of printing branch lengths, now you will use those same lengths to do some simple 'tree arithmetic'. First, compute the average distance from root to leaf within a tree.

Your function will use a codebase nearly identical to that of the previous homework, and should take the following form:

```
def avgDistance(tree):
#This function returns the average root to leaf distance
#in a phylogenetic tree
#
#...insert code here
#
# return averageDistance
```

Detecting outliers. When genes evolve faster or slower than expected, that may reveal important insight as to how natural selection is acting. Genes that are being repurposed by evolution to function in a new niche, may evolve more rapidly in that niche/species -- this process is sometimes called *positive* or Darwinian selection (note: fast rates alone are not generally sufficient to *prove* positive selection). Genes that evolve more slowly in one species may be under additional constraints or stronger *negative* selection.

In this exercise you will write two functions that detect the leaf with the shortest (and longest) distances to the root of the tree. These represent the genes that have undergone the least (and most) evolution since their divergence from a last common ancestor.

These functions should have the following forms:

```
def minDistance(tree):
#This function returns the minimum root to leaf distance
#in a phylogenetic tree, and the name of the node with the
#minimum.
#If two or more leafs are equidistant, pick the one on the left.
#
#...insert code here
#
# return [minLength, name]
```

```
def maxDistance(tree):
#This function returns the maximum root to leaf distance
#in a phylogenetic tree, and the name of the node with the
#maximum.
#If two or more leafs are equidistant, pick the one on the left.
#
#...insert code here
#
# return [maxLength, name]
```

Additional resources: A new test tree (tree3.txt) is available on the course website. Try to see if you can find the fastest and slowest evolving genes. To check your answer, try one of the following tree visualization programs:

Online:

<http://cgi-www.daimi.au.dk/cgi-chili/phyfi/go>

Download (cross-platform):

<http://darwin.zoology.gla.ac.uk/~rpage/treeviewx/>